

The calculation environment

formatting reasoned calculations and calculational proofs

Maarten Fokkinga (m.m.fokkinga@gmail.com)

January 5, 2015

Abstract

The `calculation` environment formats *reasoned calculations*, also called *calculational proofs*. The notion of reasoned calculations was originally advocated by Wim Feijen and Edsger Dijkstra. The `calculation` package accepts options `fleqn` and `leqno` (with the same effect as L^AT_EX options `fleqn` and `leqno`, and inherits these from the document class), it allows steps and expressions to be numbered (by L^AT_EX equation numbers, obeying the L^AT_EX `\label` command to refer to these numbers), and a step doesn't take vertical space if its hint is empty. An expression in a calculation can be given a comment; it is placed at the side opposite to the equation numbers.

Calculations are allowed inside hints although numbering and commenting is then disabled.

1 User manual

This package provides the `calculation` environment, to format *reasoned calculations*, also called *calculational proofs*. The *steps* in the calculation are vertically listed, and each step is accompanied by a (possibly empty) *hint*, explaining why the step is valid. This style was originally advocated by Wim Feijen and Edsger Dijkstra. The `calculation` package accepts options `fleqn` and `leqno` (with the same effect as L^AT_EX options `fleqn` and `leqno`, and inherits these from the document class), it allows steps and expressions to be numbered (obeying L^AT_EX's `\label` command to refer to these numbers). A step doesn't take vertical space if its hint is empty. An expression in a calculation can be given a comment; it is placed at the side opposite to the equation numbers.

Calculations are allowed inside hints by the `subcalculation` environment (for small calculations which do not deserve a separate discussion before or after the main calculation), although numbering and commenting is then disabled.

1.1 Example

Here is an example (with options `fleqn`, `leqno`, and `block`, and default step symbol '='); L^AT_EX source on the left, output on the right:

```

Demo:
\begin{calculation}[\approx]
  expr \comment{cmnt!}
\step*{\label{N}%
This step has nbr~\ref{N}}
  expr
\step*{}
  \label{N2}
  expr~left~part~ + ~
  \
  expr~right~part
\step[=]{
  left part:\
  \begin{subcalculation}[\leq]
    expr
  \step{hint}
    expr
  \step{hint}
    expr
  \end{subcalculation}
  \
  right part, similarly:\
  \begin{subcalculation}
    expr \step{} expr
  \step{} expr
  \end{subcalculation}
  \
  end of hint
}
\doNumber\label{E}
  this~expr~has~number~\ref{E}
\step{hint 1st line\
hint 2nd line}
  expr
\end{calculation}
Note: step (\ref{N2}) is empty.

```

```

Demo:
                                cmnt!
(1)  ≈  |  This step has nbr 1
(2)  ≈  expr
      expr left part  +
      expr right part
=     |  left part:
      expr
      ≤  |  hint
      expr
      ≤  |  hint
      expr
      right part, similarly:
      =  expr
      =  expr
      =  expr
      end of hint
(3)  this expr has number 3
      ≈  |  hint 1st line
          |  hint 2nd line
          expr
Note: step (2) is empty.

```

1.2 Usage

The calculation environment is used in its most simple form as follows:

```

\begin{calculation}
  EXPRESSION
\step{HINT}
  EXPRESSION
\step{HINT}
  EXPRESSION
  ...
\end{calculation}

```

Each HINT is normal text, each EXPRESSION is mathematical text.

Depending on various parameters, the output is, more or less, like this:

```

    EXPRESSION
=   { HINT }
    EXPRESSION
=   { HINT }
    EXPRESSION
...

```

Command `\stepsymbol` defines the default step symbol, `=` in the above example; it can be redefined with `\renewcommand`. Moreover, the `calculation` environment itself has an optional parameter, overruling the default step symbol in this specific calculation, and even each step has an optional parameter giving the step symbol for that particular step. Command `\step*` produces a step that is numbered with an equation number, as in \LaTeX ; \LaTeX command `\label` may be used to label the number so that references can be made to it. Command `\doNumber` inside an `EXPRESSION` produces an equation number. Command `\comment{TEXT}` inside an `EXPRESSION` produces `TEXT` at the side of the paper opposite to the equation number side. So, in full glory, the environment has the following appearance:

```

1 %% ===== Usage =====
2 %%
3 %%   \begin{calculation}[SYMBOL]
4 %%     EXPRESSION <-- may contain \doNumber and \comment{TEXT}
5 %%   \step*[SYMBOL]{HINT}
6 %%     EXPRESSION <-- may contain \doNumber and \comment{TEXT}
7 %%   \step*[SYMBOL]{HINT}
8 %%     EXPRESSION <-- may contain \doNumber and \comment{TEXT}
9 %%     ...
10 %%   \end{calculation}
11 %%   Each asterisk "*" and each part "[...]" is optional.
12 %%

```

An empty `HINT` takes no vertical space while the `SYMBOL` is still vertically centered in between the two `EXPRESSION`s. Both in `EXPRESSION` and in `HINT` new lines may be generated by `\\`. As in \LaTeX , `EXPRESSION` may not contain empty lines. Hints may have empty lines, even at the beginning.

Option `block` makes the entire hint (possibly several lines) into one block which, as a whole, is surrounded by delimiters.

Page breaks will occur only immediately after a step-plus-hint; a page break within or after an expression would be confusing for the reader.

1.3 Options and document style parameters

By default, calculations are placed horizontally centered on the page, but when the entire document has option `fleqn` (“flush left equations”), or when the package is given this option explicitly, then calculations are placed flush left.

By default, calculations have their step and expression numbers at the right side of the page (just as in \LaTeX), but when the entire document has option `leqno` (“left equation numbers”), or when the package is given this option explicitly, then calculations are placed at the left side of the page.

By default, the first line of a `HINT` is opened by `\Hlineopen` and the last one is closed by `\Hlineclose` (braces in the above example), but when option `block` is given to the package, then the entire hint is made into one block surrounded by `\Hblockopen` and `\Hblockclose`.

Both `\Hblockopen` and `\Hblockclose` must be “delimiters” so that they can be stretched vertically. At least the following symbols are delimiters: braces `{ }`, brackets `[]`, vertical bar `|`, vertical double bar `||`, angles `\langle \rangle`, and “nothing” (in `LATEX` indicated by a dot `.`). These commands can be redefined with `\renewcommand`.

The amount of space between `\Hlineopen` and the first token of the hint is given by `\Hsep`. The amount of indentation of the text of a hint, relative to the expressions, is given by `\Hindent` plus `\Hsep` (plus the width of `\Hblockopen` if option `block` is valid).

In summary, the following are the style parameters and options:

```

13 %% ----- DOCUMENT STYLE PARAMETERS -----
14 %% options:
15 %%   fleqn, leqno, block
16 %% commands:
17 %%   \newcommand{\stepsymbol}{=}
18 %%   \newcommand{\Hblockopen}{|}
19 %%   \newcommand{\Hblockclose}{.}
20 %%   \newcommand{\Hlineopen}{\lbrace}
21 %%   \newcommand{\Hlineclose}{\rbrace}
22 %%   \newcommand{\Hindent}{1em}
23 %%   \newcommand{\Hsep}{1ex}
24 %%   \newcommand{\calculcolsep}{\arraycolsep}
25 %%   \newcommand{\Hposv}{t}
26 %%   % vertical position of the step symbol in front of a block hint
27 %% These default values may be redefined by "\renewcommand".

```

2 The `LATEX` code for the macros

```

28
29 \def\fileversion{1.00}
30 \def\filedate{5 Jan 2015}
31 \def\docdate{5 Jan 2015}
32
33 \ProvidesPackage{calculation}[2014/12/05 Format reasoned calculations]
34

```

All auxiliary variables specific to this package are named as follows:

```

\calc@... or
\subcalc@... or
\ifcalc@... or
\endcalc@...

```

Here are the options `fleqn`, `leqno`, and `block`, and their effect:

```

35
36 \DeclareOption{fleqn}{\calc@fleqn}
37 \DeclareOption{leqno}{\calc@leqno}
38 \DeclareOption{block}{\calc@blocktrue}
39

```

2.1 Preliminary auxiliaries

To place comments and step/expr numbers at opposite sides, we place them in a very wide box at the right side of the page, and surround them by suitable fill's:

```
... \calc@eqnoLfil <exprno> \calc@eqnoRfil ...
... \calc@eqnoLfil <stepno> \calc@eqnoRfil ...
... \calc@eqnoRfil <comment> \calc@eqnoLfil ...
```

Depending on the options, these “fill”s are set to `\hfil` or `{}`. We first check that these fill commands are not yet in use:

```
40
41 \newcommand{\calc@fill}{\relax}
42 \newcommand{\calc@eqnoLfil}{\relax}
43 \newcommand{\calc@eqnoRfil}{\relax}
44
```

Eqno's at the left side means no fill at the L side and a real fill at the R side:

```
45
46 \newcommand{\calc@leqno}
47   {\def\calc@eqnoLfil{} \def\calc@eqnoRfil{\hfil}}
48
```

Eqno's at the right side:

```
49
50 \newcommand{\calc@reqno}
51   {\def\calc@eqnoRfil{} \def\calc@eqnoLfil{\hfil}}
52
```

Calculations horizontally centered on the page:

```
53
54 \newcommand{\calc@ceqn}{\def\calc@fill{fil}}
55
```

Calculations flush left: set `\calc@indent` to `\mathindent`. However, global document option `fleqn` defines `\mathindent`; if global option `fleqn` is not used, `\mathindent` is defined now:

```
56
57 \newcommand{\calc@fleqn}
58   {\@ifundefined{mathindent}
59     {\calc@indent\leftmargini}{\calc@indent\mathindent}
60     \def\calc@fill{fill}}
61
```

2.2 Default settings related to the options

By default, calculations are centered and eqno's at the right (LaTeX' default); and by default no block hints (so `\Hlineopen` at the very first line and `\Hlineclose` at the very last line):

```
62
63 \calc@ceqn
64 \calc@reqno
65 \newif\ifcalc@block \calc@blockfalse
66 \newdimen\calc@indent\calc@indent\z@skip
67
```

Recall that `\calc@indent` is set to `\mathindent` if `fleqn` is valid.

```
68
69 \ProcessOptions
70
```

2.3 Auxiliary commands of general use

```
71
72 \RequirePackage{delarray}
73
```

Package `delarray` (and hence package `array`) is required for `\left` and `\right` delimiters for array's and tabulars — which we will exploit for hints. With this package, the delimiters come out well if the array (tabular, in our case) has option `[t]` (top alignment). Command `\@ifmtarg` is my poor man's way to test for an empty argument; it should be `\protected` in moving arguments. Note: the command `\@ifmtarg` provided by `\RequirePackage{ifmtarg}` doesn't work if the argument (a HINT in our case) contains `'&'` symbols.

```
74
75 \newcommand\@ifmtarg[3]
76   {{\def\myempty{ }\def\myarg{#1}\ifx\myempty\myarg{#2}\else{#3}\fi}}
77   %% usage: \@ifmtarg {arg} {then} {else}
78
```

2.4 Default values for the document style parameters

NAMING CONVENTION: EXPR = Expression, H = Hint
--

```
79
80 \newcommand{\stepsymbol}{=}
81 \newcommand{\Hblockopen}{|}
82 \newcommand{\Hblockclose}{.}
83 \newcommand{\Hlineopen}{\lbrace}
84 \newcommand{\Hlineclose}{\rbrace}
85 \newcommand{\Hindent}{1em}
86 \newcommand{\Hsep}{1ex}
87 \newcommand{\Hposv}{t}
88 \newcommand{\calculcolsep}{\arraycolsep}
89
```

The delimiters need some preprocessing in view of the way they will be used (in command `\calc@@@step`). In particular, the spaces at the beginning of a hint will be ignored (see `\HlineOPEN` below).

```
90
91 \newcommand{\Hsepskip}{\hbox to \Hsep {}}
92 \newcommand{\HlineOPEN}
93   {\ifcalc@block\else\makebox[0pt][r]{\m@th$\Hlineopen$\Hsepskip}\fi}
94   \ignorespaces}
95 \newcommand{\HlineCLOSE}
96   {\ifcalc@block\else\makebox[0pt][l]{\m@th$\Hsepskip$\Hlineclose$\fi}
97
```

NOTE: The else clause in `\HlineCLOSE`, above, assumes that the last hint line is not empty. In order to get `\Hlineopen` and `\Hlineclose` vertically aligned in case the last hint line is empty, the else clause should read in that case:

“\makebox[0pt][r]{\m@th\$\Hlineclose\$\Hsepskip}”.

Alas, I don't know how to check “the preceding line is empty” in the definition of `\calc@@@step` below, just in front of `\HlineCLOSE`.

```
98
99 \newcommand{\HblockOPEN}{\ifcalc@block\Hblockopen\else.\fi}
100 \newcommand{\HblockCLOSE}{\ifcalc@block\Hblockclose\else.\fi}
101
```

2.5 Specific auxiliaries

Some sanity checks, and several auxiliaries:

```
102 \newcommand{\calc@origmath}{\relax}
103 \newcommand{\calc@stepsymbol}{\relax}
104 \newlength\calc@math      % to store the value of \mathsurround
105 \newif\ifcalc@emptyH     % for temporary local use only
106 \newif\ifcalc@numberedstep % true if currently in \step*
107 \newif\ifcalc@eqnoswitch  % true if an eqno will be produced
108 \newcommand{\calc@emptyHskip}{-0.5\baselineskip}% for empty hints
109 \calc@eqnoswitchfalse
```

By default the expressions in a calculation are not numbered. Whenever an expression is to be numbered, the switch is set true. This is done by `\doNumber`, which will be made available inside expressions of a calculation. Doing `\doNumber` twice should have the same effect as doing it once!

```
110 \newcommand{\calc@doNumber}
111   {\ifcalc@eqnoswitch \else
112     \global\calc@eqnoswitchtrue
113     \stepcounter{equation}
114     \gdef\@currentlabel{\p@equation\theequation}
115     \fi}
```

2.6 Main code: calculation and step

The main idea of the calculation environment is to adapt \LaTeX 's way of formatting math expressions, and `eqnarray` in particular. Thus `calculation` sets up a `\halign` with three columns:

- The 1st one for the step symbol (math mode)
- The 2nd for the hint and expression (math mode)
- The 3rd one for the comment and step/expr number (LR mode)

An eqno is placed in a large `\hbox` of length `\linewidth` which itself is considered of zero length and placed at the right in the last column. By suitable fill's the eqno then appears either at the left or right side of that large `\hbox`, and thus at the left or right side of the page.

Command `\step` will be invoked inside an expression; thus, briefly said, it should have this effect: “end the expression, begin a new line, print the step symbol and hint, and begin a new expression”.

2.6.1 The calculation environment

`calculation` The calculation environment has one optional argument, for the step symbol, the default being `\stepsymbol`; within the code for the environment, the step symbol is known as `\calc@stepsymbol`. Because the step symbol and `expr/hint` must be set with zero `mathsurround`, we set `mathsurround` to zero (by T_EX's `\m@th`), but take measures to reset `\mathsurround` to its original value inside hints.

A newline command `\\` inside expressions is delegated to `\calc@cr` (defined below).

Inside the environment, command `\step` and environment `subcalculation` are made available; outside the environment `\step` and `subcalculation` may have another meaning.

```

116
117 \newenvironment{calculation}[1][\stepsymbol]
118 {\setlength\calc@math\mathsurround
119  \def\calc@origmath{\mathsurround\calc@math}%
120  \abovedisplayskip\topsep
121  \ifvmode\advance\abovedisplayskip\partopsep\fi
122  \belowdisplayskip\abovedisplayskip
123  \belowdisplaysshortskip\abovedisplayskip
124  \abovedisplaysshortskip\abovedisplayskip
125  \def\calc@stepsymbol{#1}%
126  \tabskip\calc@indent plus 1fil
127  \let\\=\calc@cr
128  \def\step{\calc@step}% Make \step available inside calculations
129  \def\subcalculation{\calc@subcalculation}% similarly subcalculation
130  \def\endsubcalculation{\endcalc@subcalculation}%
131  $$
132  \halign to \displaywidth
133  \bgroup
134    \tabskip\z@ \hfil\m@th$ ## $\hskip\calculcolsep
135    & \m@th$ ## $\hfil\tabskip 0pt plus 1\calc@fill
136    & \hfil\llap{##}\tabskip\z@
137  \cr
138  \calc@beginEXPR
139 }

```

In the preceding line, the `\halign` declaration ends with the beginning of a math expression (`\calc@beginEXPR`, which will skip to the the 2nd column, that is, the column for expressions). The following line closes the environment; it ends the last math expression (`\calc@endEXPR`, which will step over to the last column, and print the eqno and comment):

```

140 {\calc@endEXPR
141  \egroup $$ \global\@ignoretrue\ignorespaces}
142

```

`\calc@cr` The command `\\` within expressions is delegated to `\calc@cr`; it ends the current expression, gives a little extra `vspace`, and begins a new line of the expression:

```

143
144 \newcommand\calc@cr{
145   \calc@endEXPR
146   \noalign{\nopagebreak\vskip\jot}
147   \calc@beginEXPR}
148

```

From L^AT_EX we've taken over the idea of extra `\jot` vertical space between lines of one mathematical expression. The `\nopagebreak` prevents a calculation to be split within or just after an expression; that would be too confusing for the reader.

2.6.2 The step command

`\dotsstep` Most of the work for the calculation environment, is done in command `\step` or its companions `\calc@step`, `\cal@@step` and `\calc@@@step`. Recall that `\step` has been made available inside `calculation` by a local definition that simply calls `\calc@step`. This latter one first checks whether the next token is a star `*` (and stores this information in the global `\calc@numberedstep`) and then calls `\calc@@step`. This `\calc@@step` has one optional argument, the default being `\calc@stepsymbol` (set by the calculation environment); it calls `\calc@@@step` with the step symbol:

```

149
150 \newcommand{\calc@step}
151   {\@ifstar{\global\calc@numberedsteptrue\calc@@step}
152    {\global\calc@numberedstepfalse\calc@@step}}
153
154 \newcommand{\calc@@step}[1][\calc@stepsymbol]{\calc@@@step{#1}}
155

```

`\calc@@@step` Now, the final `\calc@@@step` has two obligatory arguments:

- #1 = step symbol
- #2 = hint lines possibly including several `\`'s

As a preparation it checks whether the hint is empty and stores this in `\calc@Hempty`. Then it ends the current expression, does a negative `vskip` if the hint is empty, and increases the equation counter if the step is numbered:

```

156
157 \newcommand{\calc@@@step}[2]
158 { \@ifmtarg{#2}{\global\calc@emptyHtrue}{\global\calc@emptyHfalse}
159   \calc@endEXPR
160   \ifcalc@emptyH \noalign{\vskip \calc@emptyHskip}
161   \else \noalign{\nopagebreak\vskip\jot}
162   \fi
163   \ifcalc@numberedstep
164     \refstepcounter{equation}
165     \gdef\@currentlabel{\p@equation\theequation}
166     \gdef\calc@stepno{\theequation}
167   \fi

```

After these preparations, print the step symbol in the 1st column:

```

168   #1 &

```

Now print the hint, if not empty. First take care of the indentation, then take the hint lines as body of a tabular which has `\HblockOPEN` and `\HblockCLOSE` as delimiters and put `\HlineOPEN` in front of the 1st line and `\HlineCLOSE` after the last line. These delimiters have been defined to be null depending on the validity of option `block`. The `tabular` comes from package `delarray`, and thus has the feature of allowing the delimiters around the column specification. We also take care to restore `\mathsurround` to its original value within each hint line.

```

169 \ifcalc@emptyH
170 \else
171   \hskip\Hindent
172   \begin{tabular}[\Hposv]
173     \HblockOPEN{>{\calc@origmath\Hsepskip}l<{\Hsepskip}}\HblockCLOSE
174     \HlineOPEN #2\HlineCLOSE
175     \end{tabular}
176 \fi
177 &
178 \ifcalc@numberedstep \calc@print@theequation \fi
179 \cr

```

Now, having completed the step-and-hint line by `\cr` and before beginning the next math expression, do a negative `vskip` if the hint was empty, and put some extra space between the hint and expression (just as within expressions):

```

180 \ifcalc@emptyH \noalign{\vskip \calc@emptyHskip} \fi
181 \noalign{\vskip\jot}
182 \calc@beginEXPR
183 }
184

```

2.6.3 Remaining auxiliary commands

`\calc@beginEXPR` When `\calc@beginEXPR` is called, a new line of the `\halign` of calculation is to be filled. Since no step symbol has to be printed, we skip over to the next column (the 2nd one). This column is processed in math mode, so nothing has to be done, except for making `\doNumber` and `\comment` available and resetting the current comment to “nothing, yet”:

```

185
186 \newcommand{\calc@beginEXPR}
187   {&
188   \def\doNumber{\calc@doNumber}
189   \def\comment{\gdef\calc@comment}
190   \gdef\calc@comment{}
191   }
192

```

`\calc@endEXPR` When `\calc@endEXPR` is called, we simply step over to the last (3rd) column by `&` and print the `eqno` and `comment` (possibly null), and close the line with `\cr`:

```

193
194 \newcommand{\calc@endEXPR}
195   { & \calc@@eqno\calc@@comment \cr }
196

```

`\calc@@comment` The type setting of comments and numbers is rather straightforward: at the very right end of a wide `\hbox`, which is pretended to be of zero width, and surrounded by suitable `fill`’s to shift them to the other side, if needed.

```

197
198 \newcommand{\calc@@comment}
199   {\llap{\hbox to\linewidth
200     {\calc@eqnoRfil \normalfont\normalcolor\calc@comment
201     \calc@eqnoLfil}}}

```

```

202 \newcommand{\calc@eqno}
203   {\ifcalc@eqnoswitch
204     \calc@print@theequation
205     \global\calc@eqnoswitchfalse
206     \fi}
207 \newcommand{\calc@print@theequation}
208   {\llap{\hbox to\displaywidth
209     {\calc@eqnoLfil \m@th\normalfont\normalcolor (\theequation)%
210       \calc@eqnoRfil}}}
211

```

3 Subcalculation

There are several problems in using the `calculation` environment within hints. First, the width need be determined. (When option `fleqn` is valid, then the width is more or less the line length minus these three: `\calc@indent`, width of widest step symbol, `\Hindent`.) Second, even if the width for the sub calculation is known, it is hard to get the eqno and comment at the right place on the page. Third, in all my attempts, a sub calculation in the hint of a numbered step takes the number for it self. (This could be solved by implementing a stack or push down store in `TEX` or a kind of recursive commands that build their own stack.) In view of all this, we forbid sub calculations to have numbered steps, numbered expressions, and comments in expressions. (This seems reasonable; after all, a subcalculation within a hint should be very simple and not “view-able” from the outside.) Also, we pretend the calculation to have zero width; it is the users responsibility to observe overfull lines! Finally, as a kind of fine tuning the lay-out, we halve the indentation for the calculation, and make sub calculations flush left. Actions within `subcalculation` must not overwrite gobal variables of environment `calculation`, of course. So, we have to introduce some new global variables:

```

212
213 \newif\ifsubcalc@emptyH
214 \newcommand{\subcalc@stepsymbol}{\relax}
215
subcalculation Since numbering and comments are not allowed, we use only two columns.
216
217 \newenvironment{calc@subcalculation}[1][\stepsymbol]{%
218   \begin{minipage}[c]{0pt}
219   \abovedisplayskip 0pt
220   \belowdisplayskip \abovedisplayskip
221   \belowdisplayshortskip\belowdisplayskip
222   \abovedisplayshortskip\abovedisplayskip
223   \def\subcalc@stepsymbol{#1}%
224   \@ifundefined{mathindent}{\calc@indent\leftmargini}{}}%
225   \divide\calc@indent by 2
226   \tabskip\calc@indent plus 1fil
227   \let\=\subcalc@cr
228   \def\step{\subcalc@step}% Make \step available
229   \def\comment
230     {\@latex@error{No \string\comment\space in subcalculations}}%
231   $$
232   \m@th

```

```

233 \halign
234 \bgroup
235 \tabskip\z@ \hfil$ ## $\hskip\calcolsep
236 &
237 $ ## $\hfil\tabskip 0pt plus 1\calc@fill
238 \cr
239 &
240 \global\@ignoretrue
241 }
242 { \raisebox{0pt}[0pt][1.5ex]{}}
243 \cr \egroup $$\par\end{minipage}
244 \ignorespaces}

```

How to deal with `\\` (which will invoke `\subcalc@cr`) in hints and exprs: Close the current line (of the `\halign`) and in the next line skip the column for the step symbol:

```

245
246 \newcommand\subcalc@cr
247   {\cr
248    \noalign{\nopagebreak\vskip\jot}
249    &
250    \global\@ignoretrue\ignorespaces
251   }
252

```

`\subcalc@step` A step in a subcalculation is more or less the same as a step in the normal calculation, except that numbering is not allowed:

```

253
254 \newcommand{\subcalc@step}
255   {\@ifstar{\subcalc@@stepSTAR}{\subcalc@@step}}
256
257 \newcommand{\subcalc@@stepSTAR}[1]
258   {\@latex@error
259    {No \string\step* in subcalculations; use \string\step.}}
260
261 \newcommand{\subcalc@@step}[1][\subcalc@stepsymbol]
262   {\subcalc@@@step{#1}}
263
264 \newcommand{\subcalc@@@step}[2]
265   {\@ifmtarg{#2}
266    {\global\subcalc@emptyHtrue}{\global\subcalc@emptyHfalse}
267   \cr
268   \ifsubcalc@emptyH
269     \noalign{\vskip \calc@emptyHskip}
270   \else
271     \noalign{\nopagebreak\vskip\jot}
272   \fi
273   #1 &
274   \ifsubcalc@emptyH \else
275     \hskip\Hindent
276     \hbox to 0pt {%
277     \begin{tabular}[\Hposv]
278     \HblockOPEN{>{\calc@origmath\Hsepskip}l<{\Hsepskip}}\HblockCLOSE

```

```

279      \HlineOPEN #2\HlineCLOSE
280  \end{tabular}}
281  \fi
282  \cr
283  \ifsubcalc@emptyH \noalign{\vskip \calc@emptyHskip} \fi
284  & \global\@ignoretrue
285  }

```

Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	<code>\calc@origmath</code>	<code>\Hsep</code> 23, 86, 91
<code>\...step</code> <u>149</u> 102, 119, 173, 278	<code>\Hsepskip</code> 91, 93, 96, 173, 278
<code>\@ifmtarg</code> 75, 77, 158, 265	<code>\calc@print@theequation</code> 178, 204, 207	I
C	<code>\calc@reqno</code> 50, 64	<code>\ifcalc@block</code>
<code>\calc@@@step</code> 9, 154, 157	<code>\calc@step</code> 128, 150 65, 93, 96, 99, 100
<code>\calc@@comment</code> 195, <u>197</u>	<code>\calc@stepno</code> 166	<code>\ifcalc@emptyH</code>
<code>\calc@@eqno</code> 195, 202	<code>\calc@stepsymbol</code> 105, 160, 169, 180
<code>\calc@@step</code> 151, 152, 154 103, 125, 154	<code>\ifcalc@eqnoswitch</code>
<code>\calc@beginEXPR</code>	<code>\calc@subcalculation</code> . 129 107, 111, 203
. 138, 147, 182, <u>185</u>	calculation (environment)	<code>\ifcalc@numberedstep</code>
<code>\calc@blockfalse</code> 65 <u>116</u> 106, 163, 178
<code>\calc@blocktrue</code> 38	<code>\calculcolsep</code> 24, 88, 134, 235	<code>\ifsubcalc@emptyH</code>
<code>\calc@ceqn</code> 54, 63	<code>\comment</code> 4, 6, 8, 189, 229, 230 213, 268, 274, 283
<code>\calc@comment</code> . 189, 190, 200	D	L
<code>\calc@cr</code> 127, <u>143</u>	<code>\DeclareOption</code> 36–38	<code>\lbrace</code> 83
<code>\calc@doNumber</code> 110, 188	<code>\def</code> 188	M
<code>\calc@emptyHfalse</code> 158	<code>\doNumber</code> 4, 6, 8, 188	<code>\myarg</code> 76
<code>\calc@emptyHskip</code>	E	<code>\myempty</code> 76
. 108, 160, 180, 269, 283	<code>\endcalc@subcalculation</code> 130	N
<code>\calc@emptyHtrue</code> 158	<code>\endsubcalculation</code> . . . 130	<code>\newcommand</code> 207
<code>\calc@endEXPR</code>	environments:	P
. 140, 145, 159, <u>193</u>	calculation <u>116</u>	<code>\ProvidesPackage</code> 33
<code>\calc@eqnoLfil</code>	subcalculation <u>216</u>	R
. 42, 47, 51, 201, 209	H	<code>\rbrace</code> 84
<code>\calc@eqnoRfil</code>	<code>\HblockCLOSE</code> . 100, 173, 278	<code>\RequirePackage</code> 72
. 43, 47, 51, 200, 210	<code>\Hblockclose</code> . . . 19, 82, 100	S
<code>\calc@eqnoswitchfalse</code>	<code>\HblockOPEN</code> . . . 99, 173, 278	<code>\step</code> . 5, 7, 106, 128, 228, 259
. 109, 205	<code>\Hblockopen</code> 18, 81, 99	<code>\stepsymbol</code> 17, 80, 117, 217
<code>\calc@eqnoswitchtrue</code> . 112	<code>\Hindent</code> 22, 85, 171, 275	<code>\subcalc@@@step</code> . . . 262, 264
<code>\calc@fill</code> 41, 54, 60, 135, 237	<code>\HlineCLOSE</code> . . . 95, 174, 279	<code>\subcalc@@step</code> 255, 261
<code>\calc@fleqn</code> 36, 57	<code>\Hlineclose</code> 21, 84, 96	<code>\subcalc@stepSTAR</code> 255, 257
<code>\calc@indent</code>	<code>\HlineOPEN</code> 92, 174, 279	<code>\subcalc@cr</code> 227, 246
. 59, 66, 126, 224–226	<code>\Hlineopen</code> 20, 83, 93	
<code>\calc@leqno</code> 37, 46	<code>\Hposv</code> 25, 87, 172, 277	
<code>\calc@math</code> 104, 118, 119		
<code>\calc@numberedstepfalse</code> 152		
<code>\calc@numberedsteptrue</code> 151		

<code>\subcalc@emptyHfalse</code>	266	<code>\subcalc@step</code>	228, 253	<code>\subcalculation</code>	129
<code>\subcalc@emptyHtrue</code>	266	<code>\subcalc@stepsymbol</code>	214, 223, 261	<code>subcalculation</code> (environ- ment)	216

Change History

v1.00		v0.x	
Original idea implemented anew	4	Ugly constructed, ugly to use, nice output	4